**Построение тепловой карты:**

Import pandas

Import seaborn

Data=pandas.read\_csv(‘file.csv’)

Seaborn.heatmap(data)

**Построение столбчатой диаграммы:**

Seaborn.barplot(x=\*, y=\*)

**Построение столбчатой диаграммы конверсии по неделям:**

Import matplotlib

Import pandas

Import seaborn

Matplotlib.rcParams[‘figure.figsize’]=[15,7]

Data=pandas.read\_csv(‘file.csv’)

Conversions=data[‘payments’]/data[‘installs’]

Seaborn.barplot(x=data[‘week\_number’], y= Conversions)

**Метод Diff():**

Найти разность между текущим и предыдущим значениями столбца installs

Diff\_installs=data[‘installs’].diff()

**Индексация:**

Вывод данных по заданным условиям столбца

Import pandas

Weeks=[7,9,11]

Data=pandas.read\_csv(‘file.csv’)

Installs=data[‘installs’]

Index\_data=installs[weeks]

Print(index\_data)

**Уникальные значения и их количество столбца:**

Import pandas

Data=pandas.read\_csv(‘file.csv’)

States\_unique=data[‘State’].unique()

States\_unique\_count=data[‘State’].nunique()

Print(States\_unique)

Print(States\_unique\_count)

**Логическая индексация:**

Отобрать значения Года равные 2019 по столбцам Акры и Штаты

Import pandas

Data=pandas.read\_csv(‘file.csv’)

Acres\_2019=data[data[‘Year’]==2019][‘Acres’]

states\_2019=data[data[‘Year’]==2019][‘states’]

Print(Acres\_2019)

Print(states\_2019)

**Группировка значений:**

Группировать данные по Году и суммируем Акры каждой группы

Import pandas

Data=pandas.read\_csv(‘file.csv’)

Acres\_usa=data.groupby(‘Year’)[‘Acres’].sum()

Print(Acres\_usa)

Группировать данные по Году и считаем значения Акры в каждой группе

Acres\_usa=data.groupby(‘Year’)[‘Acres’].count()

**Отобразить индексы (порядковые номера) столбца acres\_usa:**

Acres\_usa=data.groupby(‘Year’)[‘Acres’].sum()

Years.unique=acres\_usa.index.values

**Сдвиг вперед значений в столбце:**

M=yield\_usa.shift(1) – на 1 значение

**Опция для отражения больших чисел:**

Pandas.set\_option(‘display.float\_format’, lambda x: ‘%.2f’ % x)

**Удаление первого значения столбца s методом dropna():**

s.dropna()

**Модуль столбца s:**

Abs(s)

**Посчитать значения в столбце s методом value\_counts():**

Data[‘s’].value\_counts()

**Параметры тепловой карты:**

Seaborn.heatmap(data, xticklabels=a, yticklabels=b, annot=True, cmap=’RdYlGn’)

xticklabels=a – для подписей на оси X взять значения из a

yticklabels=b– для подписей на оси Y взять значения из b

annot=True – добавить на график аннотации, то есть отображать не только цвета ячеек, но и значения

cmap=’RdYlGn’ – цветовая схема для числовой шкалы (красный – минимальные значения, зеленый – максимальные)

**Если столбец s получился вложенным из-за группировки по двум столбцам его нужно развернуть методом unstack():**

Data.groupby([‘a’,’b’])[‘c’].sum()

Data.groupby([‘a’,’b’])[‘c’].count()

s.unstack()

Пример:

import pandas  
import seaborn

data = pandas.read\_csv('support\_data.csv')

segments\_new = ['Потенциальные клиенты', 'Обычные клиенты', 'VIP-клиенты']  
intervals = ['До внедрения роботов', 'После внедрения роботов']

# для каждой пары сегмент-интервал считаем сумму продолжительностей  
duration\_sum = data.groupby(['segment', 'interval'])['duration'].sum()      
# для каждой пары сегмент-интервал считаем количество записей  
duration\_count = data.groupby(['segment', 'interval'])['duration'].count()

# считаем среднее: делим сумму на количество  
mean\_duration = duration\_sum / duration\_count  
print(mean\_duration)

# строим тепловую карту для длительности ответов  
# столбец mean\_duration получился вложенным из-за группировки по двум столбцам,  
# поэтому его нужно «развернуть» методом unstack()  
seaborn.heatmap(mean\_duration.unstack('interval'), xticklabels=intervals, yticklabels=segments\_new, annot=True, cmap='RdYlGn')